

# Recycling intermediate steps to improve Hamiltonian Monte Carlo

BY AKIHIKO NISHIMURA

*Department of Mathematics, Duke University, Durham, North Carolina 27708, U.S.A.*  
an88@duke.edu

AND DAVID B. DUNSON

*Department of Statistical Science, Duke University, Durham, North Carolina 27708, U.S.A.*  
dunson@duke.edu

## SUMMARY

Hamiltonian Monte Carlo and related algorithms have become routinely used in Bayesian computation. The utility of such approaches is highlighted in the software package STAN, which provides a platform for automatic implementation of general Bayesian models. Hence, methods for improving the efficiency of general Hamiltonian Monte Carlo algorithms can have a substantial impact on practice. We propose such a method in this article by *recycling* the intermediate leap-frog steps used in approximating the Hamiltonian trajectories. Current algorithms use only the final step, and wastefully discard all the intermediate steps. We propose a simple and provably accurate approach for using these intermediate samples, boosting the effective sample size with little programming effort and essentially no extra computational cost. We show that our recycled Hamiltonian Monte Carlo algorithm can lead to substantial gains in computational efficiency in a variety of experiments.

*Some key words:* Bayesian computation; Computational efficiency; Hamiltonian dynamics; Markov chain Monte Carlo; Metropolis-Hastings; Posterior computation.

## 1. INTRODUCTION

Markov chain Monte Carlo is routinely used for Bayesian inference, with Metropolis-Hastings providing a general subclass of algorithms that can be adapted to different settings. Many default Metropolis-Hastings algorithms are highly inefficient, and Hamiltonian Monte Carlo (Duane et al., 1987; Neal, 2010) has emerged as one of the most reliable approaches for efficient sampling in general settings. The STAN software package takes advantage of this generality and performance (Stan, 2015).

Given a parameter  $\theta \sim \pi_\theta(\cdot)$  of interest, Hamiltonian Monte Carlo introduces an auxiliary *momentum* variable  $p$  and defines a distribution  $\pi(\cdot) = \pi_\theta(\cdot) \times \mathcal{N}(0, M)$  on the augmented parameter space  $(\theta, p)$ , with  $M$  commonly referred to as the *mass matrix*. A proposal is generated by simulating trajectories of *Hamiltonian dynamics* where the evolution of the state  $(\theta, p)$  is governed by a differential equation:

$$\frac{d\theta}{dt} = -M^{-1}p, \quad \frac{dp}{dt} = -\nabla \log \pi_\theta(\theta). \quad (1)$$

Proposals generated by this mechanism can be far away from the current state and yet accepted with high probability. This behavior is due to the following property of (1): if  $\{\theta(t), p(t)\}$  denotes the solution of the differential equation with the initial condition  $\{\theta(0), p(0)\} = (\theta_0, p_0) \sim \pi(\cdot)$ , then  $\{\theta(t), p(t)\} \sim \pi(\cdot)$  for all  $t \in \mathbb{R}$ . In practice, an analytical solution to (1) is rarely available and a trajectory  $\{\theta(t), p(t)\}$  for  $0 \leq t \leq \tau$  is approximated by taking  $N \approx \tau/\epsilon$  steps of a *leap-frog* discretization scheme with stepsize  $\epsilon$ , where each step  $F_\epsilon : (\theta_0, p_0) \rightarrow (\theta_1, p_1)$  is defined via the following relations:

$$p_{1/2} - p_0 = \frac{\epsilon}{2} \nabla \log \pi_\theta(\theta_0), \quad \theta_1 - \theta_0 = \epsilon p_{1/2}, \quad p_1 - p_{1/2} = \frac{\epsilon}{2} \nabla \log \pi_\theta(\theta_1).$$

The approximate solution  $F_\epsilon^N(\theta_0, p_0) \approx \{\theta(\tau), p(\tau)\}$  no longer has the distribution  $\pi(\cdot)$ , but can be used as a Metropolis-Hastings proposal.

Current practice uses the last step  $F_\epsilon^N(\theta_0, p_0)$  as a proposal and discards all the intermediate values  $F_\epsilon^n(\theta_0, p_0)$  for  $n < N$ . As we will show, this is wasteful since the intermediate values can be *recycled* to generate additional samples from posterior distributions. The recycling algorithm only requires quantities that have already been sampled or computed, so there is essentially no extra computational cost. Our proposed recycling approach can also be applied directly to a wide variety of modified Hamiltonian Monte Carlo algorithms (Neal, 2010; Girolami and Calderhead, 2011; Fang et. al., 2014). Extensions to more complex variants are also possible as we illustrate for the No-U-Turn-Sampler (Hoffman and Gelman, 2014; Stan, 2015).

## 2. RECYCLED HAMILTONIAN MONTE CARLO

The following non-standard Hamiltonian Monte Carlo algorithm accepts or rejects each of the intermediate values, enabling recycling of these samples.

*Algorithm 1 (Recycled Hamiltonian Monte Carlo).* Generate a sequence of random variables  $\{\theta_n^{(i)}, p_n^{(i)}, n = 0, 1, \dots, N_{\max}\}_{i=1,2,\dots}$  so that the subsequence  $\{\theta_0^{(i)}, p_0^{(i)}\}_{i=1,2,\dots}$  forms a Markov chain with transition rule  $(\theta_0^{(i)}, p_0^{(i)}) \rightarrow (\theta_0^{(i+1)}, p_0^{(i+1)})$  as follows:

*Step 1.* Sample  $N^{(i)} \sim \pi_N(\cdot)$  where  $\pi_N(\cdot)$  is a distribution on  $\{1, \dots, N_{\max}\}$ .

*Step 2.* Set  $(\theta_0^{(i+1)}, p_0^{(i+1)}) = (\theta_{N^{(i)}}^{(i)}, p_{N^{(i)}}^{(i)})$  where for  $n = 1, \dots, N_{\max}$

$$(\theta_n^{(i)}, p_n^{(i)}) = \begin{cases} F_\epsilon^n(\theta_0^{(i)}, p_0^{(i)}) & \text{with probability } \min \left[ 1, \frac{\pi\{F_\epsilon^n(\theta_0^{(i)}, p_0^{(i)})\}}{\pi\{(\theta_0^{(i)}, p_0^{(i)})\}} \right] \\ (\theta_0^{(i)}, p_0^{(i)}) & \text{otherwise} \end{cases} \quad (2)$$

*Step 3.* Generate a new momentum:  $p_0^{(i+1)} \sim \mathcal{N}(0, M)$ .

Although Hamiltonian Monte Carlo discards  $(\theta_n^{(i)}, p_n^{(i)})$  for all  $n \neq 0$ , all the intermediate samples can be used as valid draws from the target distribution.

**THEOREM 1 (RECYCLING).** *If the samples  $(\theta_n^{(i)}, p_n^{(i)})$  for  $n = 1, \dots, N_{\max}$  and  $i > 0$  are generated as in Algorithm 1, then*

$$\frac{1}{KN_{\max}} \sum_{i=1}^K \sum_{n=1}^{N_{\max}} \delta_{\theta_n^{(i)}}(\cdot) \xrightarrow{w} \pi(\cdot) \text{ as } K \rightarrow \infty, \quad (3)$$

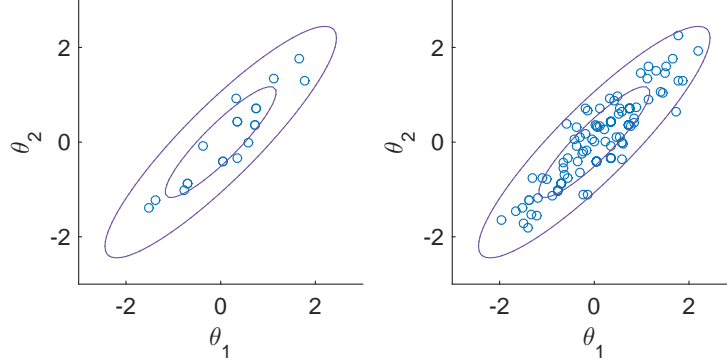


Fig. 1: Comparison of Hamiltonian Monte Carlo samples with and without recycling. The samples are drawn from a bivariate Gaussian distribution with correlation 0.9 and the contours indicate the 50% and 95% highest density region. The tuning parameters were chosen as  $\epsilon = 0.486$  and  $N^{(i)} \sim \text{Uniform}\{4, 5, 6\}$ .

where  $\xrightarrow{w}$  denotes the weak convergence of a measure.

The proof is given in the appendix.

The benefit of recycling is visually illustrated in Fig. 1. Recycling requires Metropolis-Hastings type acceptance-rejection for the intermediate steps  $F_\epsilon^n(\theta_0^{(i)}, p_0^{(i)})$  as in (2), but the calculation of acceptance probabilities typically takes little additional computational time. The unnormalized target density at the intermediate values is already computed in common variants of Hamiltonian Monte Carlo (Neal, 2010; Hoffman and Gelman, 2014) or can be obtained cheaply as a by-product of computing the gradients  $\nabla \log \pi_\theta$ .

Theorem 1 and its proof assume that, in order to recycle the intermediate values, we simulate trajectories for  $N_{\max}$  steps at each iteration of Hamiltonian Monte Carlo even if we use the  $N^{(i)}$ th leap-frog step for  $N^{(i)} < N_{\max}$  as the proposal. This is not necessary in an alternative version of the recycling algorithm described in Theorem 2. This is a minor modification, but an important one for extending the recycling idea to the No-U-Turn-Sampler. Also, the proof makes it clear that recycling is also valid when Hamiltonian Monte Carlo is used to update a subset of the parameters as part of a more complex Markov chain Monte Carlo algorithm.

**THEOREM 2 (RECYCLING).** *If the samples  $(\theta_n^{(i)}, p_n^{(i)})$  for  $n = 1, \dots, N^{(i)}$  and  $i > 0$  are generated as in Algorithm 1, then*

$$\frac{1}{\sum_{i=1}^K N_i} \sum_{i=1}^K \sum_{n=1}^{N_i} \delta_{\theta_n^{(i)}}(\cdot) \xrightarrow{w} \pi(\cdot) \text{ as } K \rightarrow \infty. \quad (4)$$

The key idea in the proof is to realize the sequence  $\{\theta_n^{(i)}, n = 1, \dots, N_i\}_{i=1,2,\dots}$  as a marginal of a Markov chain embedded in an augmented space, so that the standard ergodic theorem applied to the augmented Markov chain implies (4). The augmented Markov chain is described in the Appendix.

## 3. RECYCLED NO-U-TURN-SAMPLER

Our proposed recycling approach also applies directly to a range of modified Hamiltonian Monte Carlo algorithms (Neal, 2010; Girolami and Calderhead, 2011; Fang et. al. , 2014). An extension to a more complex proposal generation mechanism is also possible, as we illustrate with the No-U-Turns-Sampler of Hoffman and Gelman (2014). Their algorithm automates choice of path lengths by simulating each trajectory of Hamiltonian dynamics until it starts moving back towards the starting point. In order to apply such a trajectory termination criteria while preserving the reversibility of a proposal, the lengths of trajectories are recursively doubled forward or backward, checking the termination criteria at each stage of doubling.

To describe how recycling can be applied, we consider one iteration of the No-U-Turn-Sampler starting from  $(\theta_0^{(i)}, p_0^{(i)})$ . The  $n$ th doubling of a trajectory in the No-U-Turn-Sampler takes  $2^{n-1}$  leap-frog steps and produces one pair of samples  $(\theta_n^{(i)}, p_n^{(i)})$  such that the transition  $(\theta_0^{(i)}, p_0^{(i)}) \rightarrow (\theta_n^{(i)}, p_n^{(i)})$  preserves the stationary distribution. Moreover, this transition can be realized as Markovian in an augmented parameter space, analogous to Lemma 1. Thus we can recycle these intermediate values of the No-U-Turn-Sampler as valid samples from the target distribution. Formal justification of the recycled No-U-Turn-Sampler is similar to that for recycled Hamiltonian Monte Carlo in Theorem 2.

## 4. SIMULATION

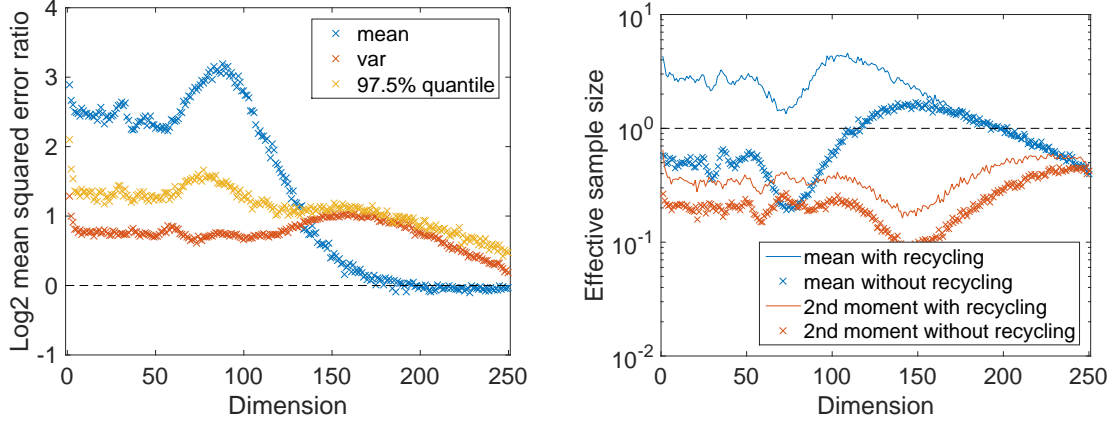
4.1. *Multivariate Gaussian*

We take our test cases from Hoffman and Gelman (2014). In all our simulations we chose the stepsizes  $\epsilon$  such that the corresponding average acceptance rates are approximately 70%, as values between 60% and 80% are typically considered optimal (Neal, 2010; Beskos et. al., 2013; Hoffman and Gelman, 2014). The dual averaging algorithm of Hoffman and Gelman (2014) was used to find such stepsizes. The choice of path lengths  $\tau^{(i)} = \epsilon N^{(i)}$  is discussed within the individual test cases below. Also, the identity mass matrix was used in all our simulations.

The first test case is sampling from a 250-dimensional multivariate Gaussian  $\mathcal{N}(0, \Sigma)$ , where  $\Sigma$  is drawn from a Wishart distribution with 250 degrees of freedom and mean equal to the identity matrix. A covariance matrix drawn from this distribution exhibits strong correlations, and in our case the ratio between the largest and smallest eigenvalue of  $\Sigma$  was approximately  $9.5 \times 10^4$ . Since Hamiltonian Monte Carlo and the No-U-Turn-Sampler with the identity mass matrix are invariant under rotations, for convenience we assume that  $\Sigma$  is diagonal with  $\Sigma_{i,i} = \sigma_i^2$ , where  $\sigma_i^2$  corresponds to the  $i$ th smallest eigenvalue of the original covariance matrix. For the path length, we first found the value  $\tau$  for which the samples in the leading principal component direction are roughly independent. The typical practice would be then to jitter  $\tau^{(i)}$ 's within the range  $[0.9\tau, 1.1\tau]$  to avoid periodicity (Neal, 2010), but this still resulted in near perfect periodicity and hence poor mixing for some parameters. After some experiments, we found jittering  $\tau^{(i)}$  in the range  $[\tau/2, \tau]$  to provide decent mixing along all the coordinates.

We simulated 1000 independent Markov chains of length 1600 starting from stationarity. We then computed the mean square error in Monte Carlo estimates of the mean, variance, and 97.5% quantile along each dimension. Fig. 2a shows  $\log_2$  of the ratios between mean square errors of Hamiltonian Monte Carlo without and with recycling. Values above zero indicate superior performance of the recycled algorithm. The ratios of mean square errors can be viewed as ratios of effective sample sizes with and without recycling. Recycling uniformly and substantially improves on estimating variance and quantiles. Though the mean estimates for parameters with

larger variances are not improved, Fig. 2b clearly demonstrates dramatic gains in the worst case performance.



(a)  $\log_2$  ratios of mean square errors without recycling (numerator) and with recycling (denominator). The  $x$ -axis corresponds to different parameters, and the horizontal line at zero to no gain for recycling.

(b) Effective sample sizes per Hamiltonian Monte Carlo step for the first and second moment estimators. The  $y$ -axis is in  $\log_{10}$  scale.

Fig. 2: Performance comparison between Hamiltonian Monte Carlo with and without recycling in estimating mean, variance, and quantiles.

We were also interested in whether recycling helps estimate the covariance structure of the target distribution. To investigate this, we computed the top eigenvalue and eigenvector of the empirical covariance matrix for each chain. We then calculated the angle between the empirical eigenvector and the plane spanned by the  $\ell$  true leading principal components. This angle should be close to 0 when the eigenvector is estimated well. To ensure identifiability of the direction, we chose  $\ell = 46$  so that  $\sigma_\ell^2 \approx \sigma_1^2/2$ , where  $\sigma_j^2$  denotes the  $j$ th largest eigenvalue. The ratios of mean squared errors in estimating the angle as well as the eigenvalues are shown in Fig. 3. We plotted the ratios against the Markov chain lengths. The direction of the principal component is not well estimated by shorter chains of lengths  $\sim 200$ , but recycling conveys a substantial advantage as the chains are run for longer.

The relative performance of the No-U-Turn-Sampler with and without recycling is similarly summarized in Fig. 4. The increase in the efficiency through recycling is more modest here since No-U-Turn-Sampler generates fewer recyclable samples compared to Hamiltonian Monte Carlo. In this test case, 9 extra samples on average were generated by recycling one iteration of No-U-Turn-Sampler while 251 extra samples were generated for Hamiltonian Monte Carlo. When a small stepsize and hence large number of leap-frog steps are necessary, one may wish to recycle every  $k > 1$  leap-frog steps to save on memory. This will likely not reduce the benefits of recycling by much since there is more and more redundancy among the recycled samples as the stepsize decreases.

#### 4.2. Hierarchical Bayesian logistic regression

The second test case is a hierarchical Bayesian logistic model regression model applied to the German credit data set available from the University of California Irvine Machine Learning Repository. Including two-way interaction terms and an intercept, we end up with 301 predictors and the regression coefficients  $\beta$  are given a prior  $\mathcal{N}(0, \sigma^2 I)$ . A hyper-prior is placed on  $\sigma^2$  in

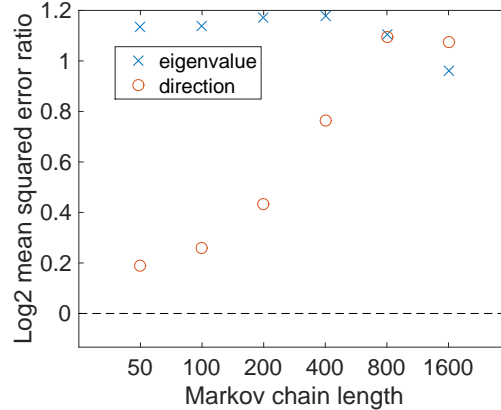


Fig. 3: Performance comparison between Hamiltonian Monte Carlo with and without recycling in estimating the direction and magnitude of the leading principal component for the covariance matrix of the target.

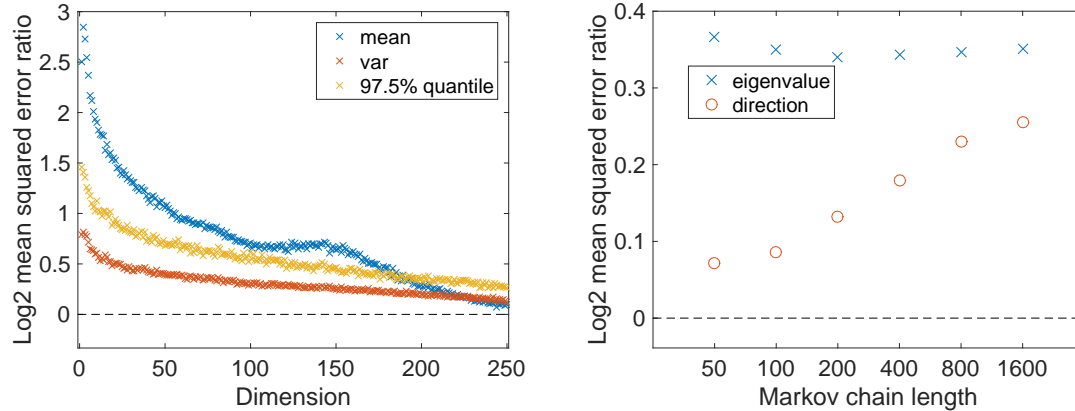


Fig. 4: Performance comparison between the No-U-Turn-Sampler with and without recycling.

order to make the posterior inference more challenging through the strong interaction between  $\sigma$  and  $\beta$ . We made one modification to the corresponding example in Hoffman and Gelman (2014) by defining our parameters to be  $\{\log(\sigma), \beta\}$  instead of  $(\sigma^2, \beta)$  since such a transformation of constrained variables has become standard (Stan, 2015). A default flat prior was placed on  $\sigma$ .

A performance comparison as in Section 4.1 is shown in Fig. 5 and 6. The 1000 independent chains were run for 3200 iterations starting from stationarity. In computing the mean squared errors, the statistics from an independent chain of  $10^7$  No-U-Turn-Sampler iterations after  $10^3$  burn-in samples were used as the ground truths. For the path lengths for Hamiltonian Monte Carlo, we first found the value  $\tau$  to maximize the normalized expected square jumping distance  $\tau^{-1/2} \mathbb{E} \|\theta^{(i+1)}(\tau) - \theta^{(i)}(\tau)\|$  as in Wang et. al. (2013), then jittered each path length  $\tau^{(i)}$  in the range  $[0.9\tau, 1.1\tau]$ . Incidentally, this optimization of path lengths can be carried out efficiently by the recycling algorithm as in Theorem 1, another advantage of recycled Hamiltonian Monte Carlo. On average, 8 extra samples were recycled for Hamiltonian Monte Carlo and 4 for the No-U-Turn-Sampler.

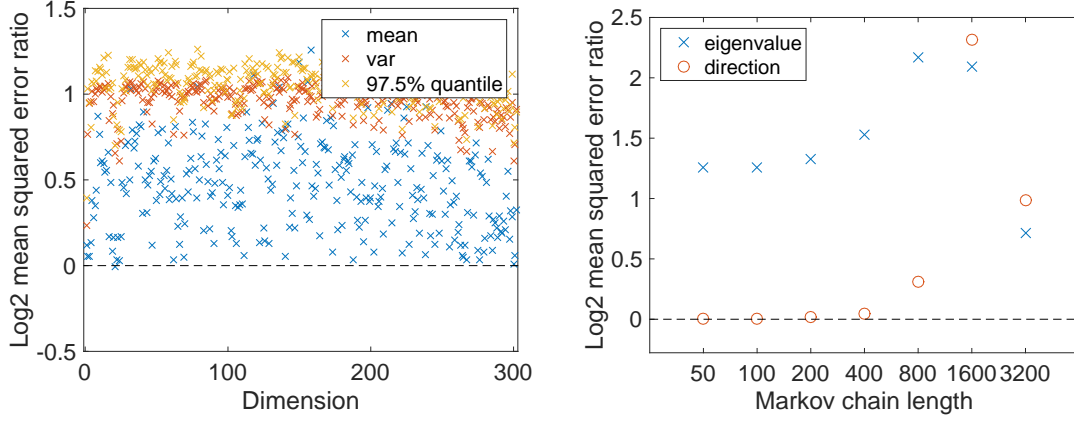


Fig. 5: Performance comparison between Hamiltonian Monte Carlo with and without recycling.

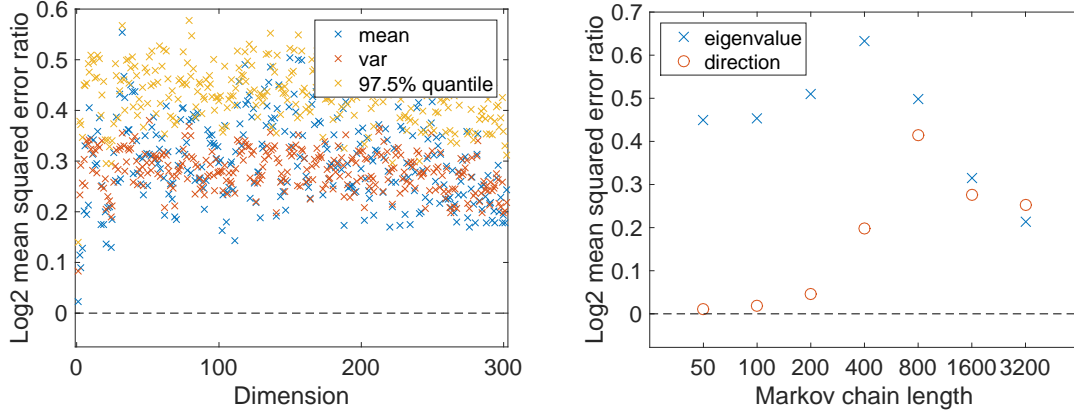


Fig. 6: Performance comparison between No-U-Turn-Sampler with and without recycling.

## APPENDIX 1

*Proofs*

*Proof of Theorem 1.* Let  $\theta^{(i)} = (\theta_0^{(i)}, \dots, \theta_{N_{\max}}^{(i)})$ . Then the sequence  $\{\theta^{(i)}\}_{i=1,2,3,\dots}$  is a Markov chain and each coordinate  $\theta_n^{(i)}$  has  $\pi(\cdot)$  as the stationary distribution. Hence, the empirical measure along each coordinate  $\theta_n^{(i)}$  of  $\theta^{(i)}$  converges to  $\pi(\cdot)$  by the standard ergodic theorem. Averaging the empirical measures along all the coordinates yields (3).  $\square$

*Proof of Theorem 2.* Theorem 2 follows immediately from the following Lemma.

LEMMA 1. Consider a Markov chain  $(\theta_0^{(i)}, p_0^{(i)}, \theta^{(i)}, p^{(i)}, n^{(i)})$  with the following update rule:

*Step 1.* Update  $(\theta_0, p_0)$ : if  $n^{(i)} = 1$ , then  $\theta_0^{(i+1)} = \theta^{(i)}$  and  $p_0^{(i+1)} \sim \mathcal{N}(0, M)$ . Otherwise,  $(\theta_0^{(i+1)}, p_0^{(i+1)}) = (\theta_0^{(i)}, p_0^{(i)})$ .

Step 2. Update  $(\theta, p)$ :

$$(\theta^{(i+1)}, p^{(i+1)}) = \begin{cases} F_{\epsilon}^{n^{(i)}}(\theta_0^{(i+1)}, p_0^{(i+1)}) & \text{with probability } \min \left[ 1, \frac{\pi \{ F_{\epsilon}^{n^{(i)}}(\theta_0^{(i+1)}, p_0^{(i+1)}) \}}{\pi \{ \theta_0^{(i+1)}, p_0^{(i+1)} \}} \right] \\ \theta_0^{(i+1)}, p_0^{(i+1)} & \text{otherwise} \end{cases}$$

Step 3. Update  $n$ : with probability  $w(n^{(i)})$ , set  $n^{(i+1)} = 1$ . Otherwise,  $n^{(i+1)} = n^{(i)} + 1$ .

Here,  $w : \mathbb{Z}^+ \rightarrow [0, 1]$  satisfies  $w(N_{\max}) = 1$  for some  $N_{\max} > 0$ . Then the stationary distribution is  $\pi(\cdot) \times \pi(\cdot) \times \pi_n(\cdot)$  where  $\pi_n$  is a discrete distribution on  $\{1, \dots, N_{\max}\}$  such that  $\pi_n(j+1) \propto \pi_n(1) \prod_{i=1}^j \{1 - w(i)\}$ .

It is not difficult to verify that the distribution described in Lemma 1 is indeed the stationary distribution of the Markov chain, and this completes the proof.  $\square$

#### REFERENCES

- Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J. M., & Stuart, A. (2013) Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, **19**, 1501–1534.
- Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987) Hybrid Monte Carlo. *Physics Letters B*, **195**, 216–222.
- Fang, Y., Sanz-Serna, J. M., & Skeel, R. D. (2014) Compressible generalized hybrid Monte Carlo. *Journal of Chemical Physics*, **140**, 174108.
- Girolami, M. & Calderhead, B. (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Statist. Soc. B*, **73**, 123 – 214.
- Hoffman, M. D. & Gelman, A. (2014) The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, **15**, 1593–1623.
- Neal, R. M. (2010) MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*. London: CRC Press.
- Stan Development Team. (2015) *Stan Modeling Language Users Guide and Reference Manual, Version 2.8.0.*, <http://mc-stan.org/>.
- Wang, Z., Mohamed, S., & Freitas, N. D. (2013) Adaptive Hamiltonian and Riemann Manifold Monte Carlo samplers. In *Proceedings of the 30th International Conference on Machine Learning*, **28**, 1462 – 1470.